Hindawi
Security and Communication Networks
Volume 2017, Article ID 3075210, 16 pages
https://doi.org/10.1155/2017/3075210

WILEY | Hindawi

Research Article

Design and Implementation of a Mobile Voting System Using a Novel Oblivious and Proxy Signature

Shin-Yan Chiou,[1,2,3] Tsung-Ju Wang,[1] and Jiun-Ming Chen[1]

[1]*Department of Electrical Engineering, College of Engineering, Chang Gung University, 259 Wen-Hwa 1st Road, Kweishan, Taoyuan, Taiwan*
[2]*Department of Nuclear Medicine, Linkou Chang Gung Memorial Hospital, Taoyuan, Taiwan*
[3]*Center for Biomedical Engineering, Chang Gung University, Taoyuan, Taiwan*

Correspondence should be addressed to Shin-Yan Chiou; ansel@mail.cgu.edu.tw

Received 2 August 2017; Revised 18 October 2017; Accepted 31 October 2017; Published 24 December 2017

Academic Editor: Georgios Kambourakis

Electronic voting systems can make the voting process much more convenient. However, in such systems, if a server signs blank votes before users vote, it may cause undue multivoting. Furthermore, if users vote before the signing of the server, voting information will be leaked to the server and may be compromised. Blind signatures could be used to prevent leaking voting information from the server; however, malicious users could produce noncandidate signatures for illegal usage at that time or in the future. To overcome these problems, this paper proposes a novel oblivious signature scheme with a proxy signature function to satisfy security requirements such as information protection, personal privacy, and message verification and to ensure that no one can cheat other users (including the server). We propose an electronic voting system based on the proposed oblivious and proxy signature scheme and implement this scheme in a smartphone application to allow users to vote securely and conveniently. Security analyses and performance comparisons are provided to show the capability and efficiency of the proposed scheme.

1. Introduction

In recent years, network transactions for applications such as Internet auctions and banking have increased greatly. Network and mobile security technologies [1–6] play important roles in protecting users' privacy. In this regard, digital signatures have attracted considerable attention. By using public-key cryptography, a signer can sign a message using his or her private key, which is owned only by the signer, to create a digital signature for the message. Then, any verifier can validate the correctness of this signature by using the signer's public key.

However, it is necessary to protect the privacy of signature receivers in some situations, such as the contents of signed message in a digital cash system or the choices from candidates in an e-voting situation. In 1983, Chaum [7] introduced a blind signature scheme to offer blindness which protects the signee's privacy. In 2013, Nayak et al. [8] proposed a blind signature scheme based on an elliptic curve discrete logarithm problem. In 2005, Rabin [9] introduced the concept of oblivious transfer. In 1994, Chen [10] proposed the concept of oblivious signatures and considered two types of oblivious signature schemes. In 2008, Tso et al. [11] provided formal definitions and security requirements for an oblivious signature scheme. In 2012, Chou [12] proposed a more efficient and secure k-out-of-n oblivious transfer scheme.

In 1996, Mambo et al. [13] proposed the concept of proxy signature. Various proxy signature schemes have been proposed [14, 15]. In 2000, Lin and Jan [16] proposed the first proxy blind signature scheme that combines the functionalities of both proxy signatures and blind signatures. In 2002, Tan et al. [17] proposed a proxy blind signature scheme; however, in 2003, Lal and Awasthi [18] showed this scheme to be insecure and further proposed a new scheme that is secure and more efficient than Tan et al.'s scheme. In 2013, Yang and Liang [19] proposed a new proxy blind signature scheme that allows revocation.

For electronic voting systems, in 2001, Ray and Narasimhamurthi [20] introduced an online anonymous electronic

voting protocol that allows a voter to cast his or her ballot anonymously by exchanging untraceable authentic messages. In 2013, Pan et al. [21] proposed an electronic voting scheme that is based on the ring signature and is resistant to a clash attack. Several schemes with delegated voting functionality have been proposed. In 2013, Zwattendorfer et al. [22] proposed a proxy voting scheme that allows a voter to delegate his or her voting power to a proxy who actually casts the ballots for all represented voters. Norway has used an Internet-based voting protocol for some years, and the vote privacy and correctness of this scheme have been demonstrated [23]. In 2016, Kulyk et al. [24] proposed a new coercion-resistant proxy voting scheme by extending the coercion-resistant JCJ/Civitas theme, aiming to prevent direct voter coercion, delegation coercion, and proxy coercion. They also proposed a new proxy voting scheme [25] to extend the Helios voting system [26] with delegated voting functionality. In 2017, Cohensius et al. [27] considered a social choice problem and demonstrated that the mechanism using proxy voting better approximates the optimal outcome.

1.1. Motivation. Compared with a blind signature scheme, a oblivious signature scheme used in e-voting provides one more property: ambiguity in selected messages. A signer cannot find out which message a voter has selected while signing the messages, but the signer can be certain that the message the voter chooses is one of the predetermined messages; otherwise, the signature would not be accepted by a verifier. Therefore, in oblivious signature systems, which differ from blind signature schemes, the limited signed contents can prevent potential malicious users from obtaining valid signatures of some candidates for unauthorized purposes.

In addition, because each unit of a group (such as each state of a country, each county of a state, each campus of a school, or each approved bank of a group) may use different methods to authorize their members (using different keys), polling booths with proxy ability are required. Additional benefits include reducing the load at voting centers and avoiding network jams. Moreover, the mobility of the voting functionality allows people to vote from anywhere using their mobile devices, thereby making the electronic voting system more convenient.

The goal in this research is a design of novel schemes which combine oblivious and proxy signatures and extend the designed schemes to an electronic voting system which provides the following properties: mobility, instant voting, proxy signer, completeness, unforgeability, unlinkability, undeniability, accuracy, distinguishability, ambiguity, nonduplication, eligibility, verifiability, fairness, and privacy, where fairness means that no one can know the current total number of votes received by every candidate before the end of the voting period.

1.2. Our Contribution. In this paper, based on the Schnorr signature [28], we propose two novel 1-out-of-n blind (oblivious) and proxy signature schemes that combine the advantages of oblivious signatures and proxy signatures and satisfy the security properties of these two signature schemes. One of the proposed proxy oblivious schemes is of the proxy-unprotected type and the other is of the proxy-protected type. Based on our schemes, we also propose an anonymous electronic voting system with proxy signer. By using the concept proposed in [29], we conduct security analyses and performance comparisons. The results showed that our scheme has good performance and is efficient. Finally, we implement the voting system on Android mobile phones to prove that our scheme is workable. This paper extends the research [30], which presents the concept of 1-out-of-n oblivious and proxy signature schemes of proxy-unprotected type without the voting system application, security analyses and formal proofs, and mobile phone implementation.

2. Related Works

In this section, we present two representative protocols that are relevant to our scheme: oblivious signature and proxy signature.

2.1. Oblivious Signature. In 1983, Chaum [7] introduced a blind signature scheme. Compared with a normal signature, a blind signature offers an additional property, blindness, that provides it with the ability to protect the signee's privacy. In a blind signature scheme, a signee could get a message's digital signature signed by a signer without revealing any information about the message. This is vital in some applications such as electronic payment systems and secure voting systems [31–35], because the requester's messages may be sensitive. Nayak et al. [8] also proposed a blind signature scheme based on an elliptic curve discrete logarithm problem.

In 2005, Rabin [9] introduced the concept of oblivious transfer. In this protocol, a sender sends some subsets of some messages but does not know what the receiver has received. Thus, the receiver can get the particular message he or she wants without revealing any information about the message to the sender. In 2012, Chou [12] proposed a more efficient and secure k-out-of-n oblivious transfer scheme.

In 1994, Chen [10] proposed the concept of oblivious signatures. He considered two types of oblivious signature schemes. The first one comprises n keys and one message; the receiver can get a message signed with one of n keys that are chosen by him or her while the signers cannot know which key has been used for the signature by the receiver. The second one comprises n messages and one key; a signee can choose one predetermined message to get signed while not revealing any information about the selected message to the signer. In contrast to blind signatures, oblivious signatures can guarantee that the signed message is actually one of the predetermined messages; therefore, if the receiver were to submit some additional messages, the signature would not be accepted by the scheme.

In 2008, Tso et al. [11] noted that Chen's proposal does not crisply formalize the notion and security properties of the scheme. Consequently, they provided formal definitions and security requirements for an oblivious signature scheme, including completeness, unforgeability, and ambiguity, and proposed a 1-out-of-n oblivious signature agreement based on Schnorr's blind signature [28]. They also improved the scheme's performance. The preceding properties render

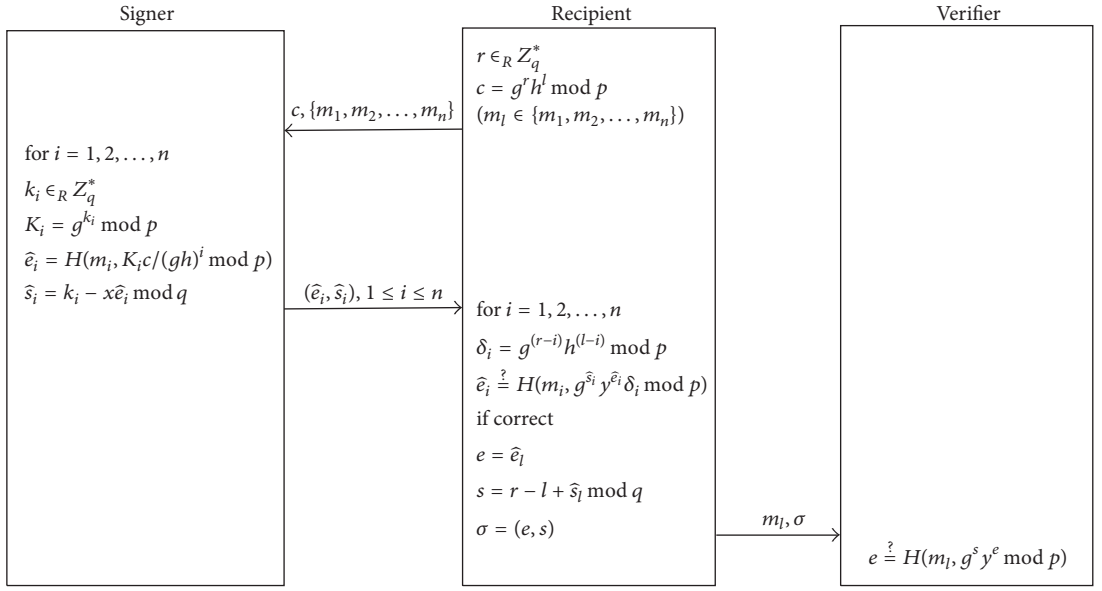


Figure 1: Oblivious signature scheme.

oblivious signatures very suitable for electronic voting applications.

In their scheme [11], it first creates a normalized definition for the oblivious signature agreement and proposes the following properties to ensure security:

(1) Completeness: as long as the recipient and the signer can implement the agreement honestly, once the agreement is completed the recipient can obtain the signed message.

(2) Unforgeability: despite the algorithm being publicly published, attackers still have difficulty creating a forged signature within an acceptable time frame.

(3) Ambiguity in selected messages: the signer is unable to determine the recipient's selection.

This system provides for three roles, namely, the signer, the recipient, and the verifier. The operation proceeds through three phases, namely, initiation, signing, and verification. Figure 1 shows their scheme and the steps in this scheme are as follows.

(1) System Initiation Phase. In this phase, the signer generates public parameters and a pair of asymmetric keys for use in the following process.

The signer first establishes security parameter 1^k to input into the system to establish the algorithm $\mathcal{G}$, which can be used to obtain the public parameters for the agreement.

(A) p, q are two large prime numbers, and $q \mid (p-1)$.

(B) $g, h \in_R Z_p^*$ are two parameters with order q.

(C) $H()$ is a hash function.

The signer also selects a random number $x \in_R Z_q^*$ as a private key to calculate the public key $y = g^x \bmod p$.

(2) Signing Phase. This phase explains how the recipient obtains the signature value from the signer's signature (see Figure 1).

Step 1. Assume the recipient wants to obtain the signature value for message $m_l \in \{m_1, m_2, \ldots, m_n\}$. First, the recipient selects $r \in_R Z_q^*$, then calculates $c = g^r h^l \bmod p$, and then transmits c and $\{m_1, m_2, \ldots, m_n\}$ to the signer.

Step 2. After the signer receives these, he selects random numbers $k_i \in_R Z_q{}^*$, $i = 1, 2, \ldots, n$, using k_i to calculate $K_i = g^{k_i} \bmod p$ followed by $\hat{e}_i = H(m_i, K_i c/(gh)^i \bmod p)$ and uses private key x to obtain $\hat{s}_i = k_i - x\hat{e}_i \bmod q$ for the signed message. Finally, $(\hat{e}_i, \hat{s}_i)$, $i = 1, 2, \ldots, n$, is transmitted to the recipient.

Step 3. Once the recipient has received $(\hat{e}_i, \hat{s}_i)$, he/she then calculates $\delta_i = g^{(r-i)} h^{(l-i)} \bmod p$, $i = 1, 2, \ldots, n$, and verifies $(\hat{e}_i, \hat{s}_i)$ to determine whether it satisfies $\hat{e}_i \stackrel{?}{=} H(m_i, g^{\hat{s}_i} y^{\hat{e}_i} \delta_i \bmod p)$, where only the lth message initially selected by the recipient will be successfully verified.

Step 4. The successful verification $(\hat{e}_l, \hat{s}_l)$ is then used to calculate $e = \hat{e}_l$ and $s = r - l + \hat{s}_l \bmod q$. Finally, we obtain the m_l's signature value $\sigma(m_l) = (e, s)$.

(3) Verification Phase. This phase explains how the verifier verifies the correctness of the signature obtained by the recipient (see Figure 3).

Step 1. The recipient sends the obtained signature value $\sigma(m_l)$ and the selection m_l to the verifier.

Step 2. After the verifier obtains $(m_l, \sigma(m_l))$, he/she determines whether $(m_l, \sigma(m_l))$ satisfies $e \stackrel{?}{=} H(m_l, g^s y^e \bmod p)$.

If it does, this represents that the signature obtained by the recipient is genuine.

Note that, in Step 2 of signing phase, the signer cannot learn which message is selected by the recipient; this is called the ambiguity of the protocol. Moreover, in Step 2 of verification phase, unless the selected message m_l is one of the elements of m_i, the signature will not be accepted by a verifier. This protocol ensures that a signee's message is one of the predetermined messages, which is one of the features of an oblivious signature scheme.

2.2. Proxy Signature. Although the aforementioned signature schemes seem practical, they are unsuitable for applications in which the signers are not always available. To overcome this problem, in 1996, Mambo et al. [13] proposed the concept of proxy signature. Various proxy signature schemes have been proposed [14, 15]. Such schemes consist of three entities: an original signer, a proxy signer, and a signee. The original signer can delegate his or her signing power to one or more proxy signers to enable them to sign messages submitted on his or her behalf. Mambo et al. discussed two types of proxy signature schemes: proxy unprotected and proxy protected.

Proxy-unprotected proxy signatures include two cases: full delegation and partial delegation. In full delegation, the original signer gives his or her private key to the proxy signer to sign messages; therefore, the original signer takes full responsibility for the signatures produced by the proxy signer. In partial delegation, the original signer creates a proxy signature key using his or her private key and securely forwards it to the proxy signer; then, the proxy signer uses this proxy signature key to sign messages on the original signer's behalf.

Proxy-protected proxy signature schemes allow the original signer to use his or her private key to create a proxy signature key and securely forward it to the proxy signer. The proxy signer computes a new proxy signature signing key—which contains both the original signer's original proxy signature signing key and the private key of the proxy signer—and then signs messages using the new proxy signature key. Therefore, the original signer and proxy signer share the responsibility for the valid proxy signatures generated by the proxy signer. Both of these schemes afford security properties such as verifiability, unforgeability, and undeniability.

For example, the president of a company could ask his or her reliable secretary (using the proxy-unprotected signature scheme) or employee (using the proxy-protected signature scheme) to sign some important documents on his or her behalf. Then, if some illegal situation arises in relation to the proxy signature, the employee's privacy can be uncovered while the secretary's privacy cannot.

Moreover, delegation can be categorized as full delegation, partial delegation, and delegation by warrant, as follows:

(1) Full delegation: the proxy signer obtains a copy of the original signer's signature key to produce a proxy signature value identical to the signature of the original signer.

(2) Partial delegation: the proxy signer's signature key is obtained through a calculation based on the original signer's private key. However, the proxy signature key cannot be used to obtain information related to the original signer's private key. Partial delegation can be categorized as one of two types: proxy-unprotected or proxy-protected. In the former, the original signer and proxy signer can both provide valid proxy signatures. In the latter, only the proxy signer can provide a valid proxy signature.

(3) Delegation by warrant: a warrant based on the original signer's signature is used to validate the proxy signer's signing authority. The proxy signer's authorization message and proxy signature content is included in the proxy signature and the verifier is used to determine the legitimacy of the authorization.

The original signer first delegates his/her signing rights to one or more proxy signers, so that the proxy signers can sign a message in the name of the original signer. Each type of the scheme, proxy unprotected or proxy protected, has to satisfy three security requirements.

(1) Verifiability: the proxy signature created by the proxy signer can convince anyone of the permission from the original signer.

(2) Unforgeability: only the authorized proxy signers can sign a valid signature; this cannot be done even by the original signer.

(3) Undeniability: neither the proxy signer nor the original signer can deny the signature they created after the signature creation.

The earliest proxy signature scheme (Figure 2) (proposed by Mambo et al. [13]) is a proxy signature of the proxy-unprotected type for the ElGamal scheme [36]. The steps in this scheme are as follows.

(1) Key Generation Phase. In this phase, the system generates two primes and a pair of keys, public and private keys, for use in the following process.

Step 1. Choose two large prime numbers p, q such that $q \mid (p-1)$.

Step 2. Select random numbers $g \in Z_p^*$ such that $\mathrm{Ord}_p g = q$.

Step 3. Each of the original signer and the proxy signer choose a random number $x \in_R Z_q^*$ as their private keys.

Step 4. Each of the original signer and the proxy signer compute $y = g^x \bmod p$ as their public keys.

(2) Proxy Phase

Step 1. The original signer chooses a random number $k \in_R Z_{p-1}^*$.

Step 2. The original signer computes $K = g^k \bmod p$.

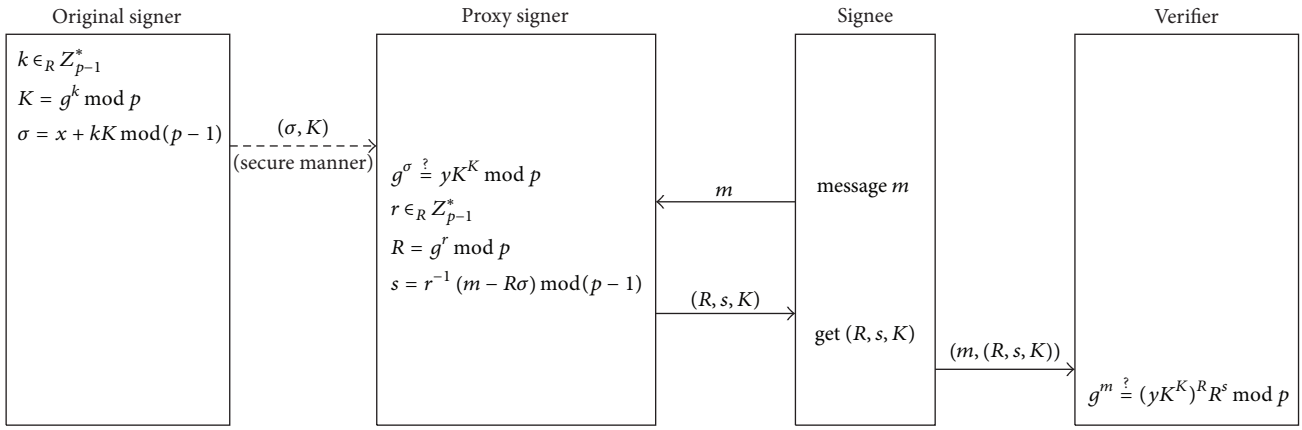


Figure 2: Proxy signature scheme.

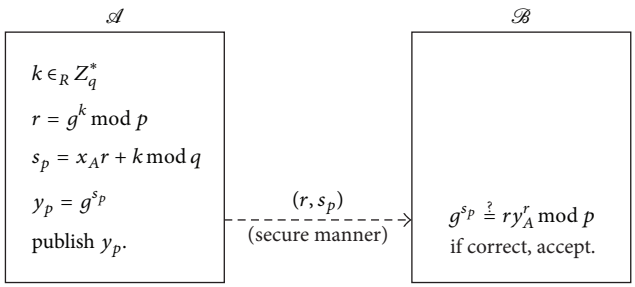


Figure 3: Proxy phase (proxy-unprotected type).

Step 3. The original signer calculates $\sigma = x + kK \bmod (p-1)$ and sends σ to the proxy signer via a secure channel.

Step 4. After receiving σ, the proxy signer checks whether $g^\sigma \stackrel{?}{=} yK^K \bmod p$ is correct. If it is, the proxy signer accepts delegation.

(3) Signing Phase. After receiving a message m from a requestor, the proxy signer processes the follow steps.

Step 1. Choose a random number $r \in_R Z^*_{p-1}$.

Step 2. Compute $R = g^r \bmod p$.

Step 3. Evaluate $s = r^{-1}(m - R\sigma) \bmod (p-1)$ and send signature (R, s, K) to the requestor.

(4) Verification Phase. After receiving a message $(m, (R, s, K))$ from the requestor, a verifier checks the validation of the signature via the following verification equation: $g^m \stackrel{?}{=} (yK^K)^R R^s \bmod p$.

Note that $(yK^K)^R R^s = (g^x g^{kK})^R g^{rs} = g^{R(x+kK)} g^{r[r^{-1}(m-R\sigma)]} = g^{R(x+kK)+(m-R\sigma)} = g^{R(x+kK)+m-R(x+kK)} = g^m \bmod p$.

3. System Goal and Security Requirements

In this section, we define the attacker model, propose two new signature schemes, and present one electronic voting system based on the proposed signature schemes.

3.1. Attacker Model for Signature Scheme. The proposed signature schemes consist of four entities: an original signer $\mathscr{A}$, a proxy signer $\mathscr{B}$, a receiver $\mathscr{R}$, and a verifier $\mathscr{V}$. In our scheme, we assume the channels between $\mathscr{A}$ and $\mathscr{B}$ are secure. Any identity (i.e., $\mathscr{R}$ or $\mathscr{V}$) communicates with $\mathscr{B}$ via an insecure public channel, offering adversaries opportunities to intercept. In the following, we present the assumptions of the attacker model [37, 38].

(1) An adversary may eavesdrop on all communications between protocol actors over the public channel.

(2) An attacker can modify, delete, resend, and reroute the eavesdropped message.

(3) An attacker cannot intercept a message over a secure channel.

(4) An attacker cannot be a legitimate original signer or proxy signer.

(5) The attacker knows the protocol description, which means the protocol is public.

3.2. Signature Scheme. We define the security properties of a proxy oblivious signature scheme as follows.

Definition 1 (security requirements of our signature scheme). Our signature scheme is secure if it achieves (1) completeness, (2) unforgeability, (3) unlinkability, (4) undeniability, (5) verifiability, (6) distinguishability, and (7) ambiguity.

The security requirements of our proposed scheme are listed as follows:

(1) Completeness: if all entities follow the protocol honestly, then, at the end of the protocol, $\mathscr{R}$ will obtain the valid signature σ of the selected message.

(2) Unforgeability: in a proxy-unprotected type scheme, $\mathscr{B}$ can sign messages on behalf of $\mathscr{A}$ without taking responsibility for the signature. $\mathscr{V}$ can merely know that the signature is signed by some proxy delegated by $\mathscr{A}$, and no one except $\mathscr{B}$ (or $\mathscr{A}$) can produce a valid signature. In a proxy-protected type scheme, $\mathscr{A}$ and $\mathscr{B}$ share the responsibility for the signature, and only $\mathscr{B}$ can create a valid signature for $\mathscr{R}$. Consequently, the proxy signing key is practically unbreakable, making it almost impossible for an attacker to create a valid signature.

(3) Unlinkability: $\mathscr{B}$ can identify neither the message nor the proxy signature he or she generates associated with the scheme after the signature is revealed when necessary.

(4) Undeniability: neither $\mathscr{A}$ nor $\mathscr{B}$ can deny the signature they have created after signature generation.

(5) Verifiability: the signature that $\mathscr{R}$ receives should be able to convince $\mathscr{V}$ of the agreement from $\mathscr{A}$ and $\mathscr{B}$.

(6) Distinguishability: the proxy signature is distinguishable from a normal one.

(7) Ambiguity: $\mathscr{B}$ cannot find out which message $\mathscr{R}$ has selected while signing the messages, but $\mathscr{B}$ can be certain that the message he or she signs is one of the predetermined messages; otherwise, the signature would not be accepted by a verifier.

3.3. Attacker Model for Voting Scheme. The proposed electronic voting system consists of five entities: a creator $\mathscr{A}$, a proxy creator $\mathscr{B}$, a voter $\mathscr{R}$, a voting center $\mathscr{V}$, and a bulletin board $\mathscr{BB}$. In our scheme, we assume the channels between $\mathscr{A}$ and $\mathscr{B}$ are secure. Any identity (i.e., $\mathscr{R}$, $\mathscr{V}$, or $\mathscr{BB}$) communicates with $\mathscr{B}$ via an insecure public channel, offering adversaries opportunities to intercept. In the following, we present the assumptions of the attacker model [37, 38].

(1) An adversary may eavesdrop on all communications between protocol actors over the public channel.

(2) An attacker can modify, delete, resend, and reroute the eavesdropped message.

(3) An attacker cannot intercept a message over a secure channel.

(4) An attacker cannot be a legitimate creator or proxy creator.

(5) The attacker knows the protocol description, which means the protocol is public.

3.4. Voting System. The proposed electronic voting system consists of five entities, $\mathscr{A}$, $\mathscr{B}$, $\mathscr{R}$, $\mathscr{V}$, and $\mathscr{BB}$, which, in a practical situation, would correspond to a central government, a local government, a voter, a voting center computer, and a bulletin website, respectively. Anane et al. [39] present the core properties of the voting system and indicated that a viable e-voting system should possess (1) accuracy, (2) privacy (untraceability), (3) individual and universal verifiability, (4) eligibility, and (5) fairness. Based on [39], we define the system requirements of the proposed electronic voting system as follows.

Definition 2 (system requirements of our electronic voting system). Our electronic voting system is secure if it achieves (1) proxy completeness, (2) eligibility, (3) nonduplication, (4) fairness, (5) accuracy, (6) verifiability, and (7) privacy.

This system requirements of our electronic voting system are listed as follows:

(1) Proxy completeness: the proxy creator should not deny the fact that he or she has accepted the delegation from the creator.

(2) Eligibility: only voters who have the right to vote can participate in the voting event.

(3) Nonduplication: every legal voter is allowed to cast only one ballot.

(4) Fairness: before the end of the voting period, no one can know the current total number of votes received by every candidate.

(5) Accuracy: the validity of all ballots can be verified by using the proper public key.

(6) Verifiability: every voter should be able to confirm his or her voting condition and verify all other voters' ballots.

(7) Privacy: no one except for the voter can find out which candidate has been chosen, even at the end of voting.

4. Proposed Signature Scheme

Two types of proxy oblivious signature schemes are proposed: proxy-unprotected type and proxy-protected type. Each scheme includes four phases: (1) system setup phase, (2) proxy phase (Figure 3), (3) signing phase (Figure 4), and (4) verification phase (Figure 5). Notation illustrates the notations used in the protocol.

4.1. Proxy-Unprotected Type. For a proxy-unprotected type scheme, our protocol is as follows.

(1) System Setup Phase

Step 1. Two large primes $p, q \ni q \mid (p-1)$ are chosen.

Step 2. Two generators $g, h \in Z_p^*$, where $\mathrm{Ord}_p g = q$ and $\mathrm{Ord}_p h = q$, are chosen.

Step 3. The original signer $\mathscr{A}$ chooses $x_A \in Z_q^*$ and computes $y_A = g^{x_A} \bmod p$.

Step 4. The proxy signer $\mathscr{B}$ chooses $x_B \in Z_q^*$ and computes $y_B = g^{x_B} \bmod p$.

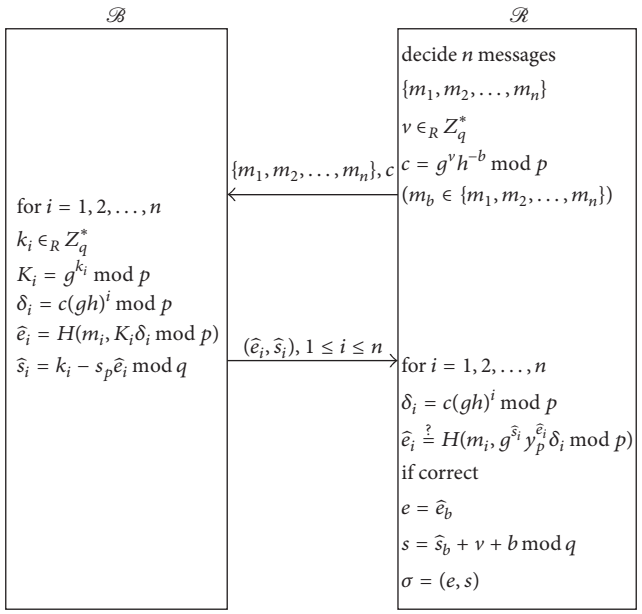


Figure 4: Signing phase.

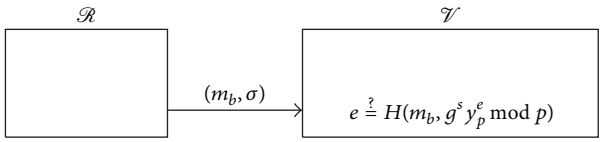


Figure 5: Verification phase.

(2) Proxy Phase

Step 1 (commission generation). $\mathscr{A}$ randomly chooses $k \in_R Z_q^*$ and computes $r = g^k \bmod p$, $s_p = x_A r + k \bmod q$, and $y_p = g^{s_p} \bmod p$.

Step 2 (proxy delivery). $\mathscr{A}$ securely forwards the pair (r, s_p) to the proxy signer $\mathscr{B}$ and publishes y_p.

Step 3 (proxy verification). $\mathscr{B}$ checks whether $g^{s_p} = r y_A^r \bmod p$ holds. If it does, $\mathscr{B}$ accepts the proxy and uses s_p as his or her secret proxy signature key.

(3) Signing Phase

Step 1. $\mathscr{R}$ decides n messages $\{m_1, m_2, \ldots, m_n\}$ and then selects a message $m_b \in \{m_1, m_2, \ldots, m_n\}$, randomly chooses $v \in_R Z_q^*$, computes $c = g^v h^{-b} \bmod p$, and finally sends the determined messages and c to $\mathscr{B}$.

Step 2. For $i = 1, 2, \ldots, n$, $\mathscr{B}$ chooses n random numbers $k_i \in_R Z_q^*$, computes $K_i = g^{k_i} \bmod p$, $\delta_i = c(gh)^i \bmod p$, $\widehat{e}_i = H(m_i, K_i \delta_i \bmod p)$, and $\widehat{s}_i = k_i - s_p \widehat{e}_i \bmod q$, and sends $(\widehat{e}_i, \widehat{s}_i)$ to $\mathscr{R}$, where $1 \le i \le n$.

Step 3. For $i = 1, 2, \ldots, n$, $\mathscr{R}$ computes $\delta_i = c(gh)^i \bmod p$ and accepts the oblivious signature if and only if $\widehat{e}_i = H(m_i, g^{\widehat{s}_i} y_p^{\widehat{e}_i} \delta_i \bmod p)$.

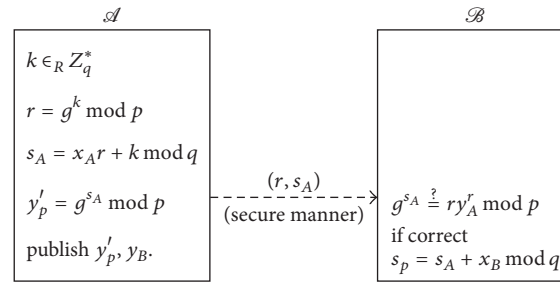


Figure 6: Proxy phase (proxy-protected type).

Step 4. To convert the oblivious signature into a generic signature, $\mathscr{R}$ lets $e = \widehat{e_b}$ and computes $s = \widehat{s_b} + v + b \bmod q$. The signature for m_b is $\sigma = (e, s)$.

(4) Verification Phase. As shown in Figure 5, the verifier $\mathscr{V}$ accepts the signature σ as a valid signature if and only if $e = H(m_b, g^s y_p^e \bmod p)$.

4.2. Proxy-Protected Type. For a proxy-protected type scheme, the signing phase and verification phase are the same as those in the case of a proxy-unprotected type scheme except for one more computation $y_p = y_p' y_B \bmod p$ (as shown in Figure 6). The proxy phase is modified as described subsequently.

(1) Proxy Phase

Step 1 (commission generation). $\mathscr{A}$ randomly chooses $k \in_R Z_q^*$ and computes $r = g^k \bmod p$, $s_A = x_A r + k \bmod q$, and $y_p' = g^{s_A} \bmod p$.

Step 2 (proxy delivery). $\mathscr{A}$ securely forwards the pair (r, s_A) to the proxy signer $\mathscr{B}$ and publishes y_p'.

Step 3 (proxy verification). $\mathscr{B}$ checks whether $g^{s_A} = r y_A^r \bmod p$ holds. If it does, $\mathscr{B}$ accepts the proxy and computes $s_p = s_A + x_B \bmod q$ as his or her secret proxy signature key.

5. Proposed Voting System

5.1. System Overview. Based on the proposed signature scheme, our electronic voting system allows a creator (central government) to delegate one or more proxy creators (local government), and a voter can get a legal ballot from a proxy creator and send his or her vote to a verifier (center voting computer).

This system includes six phases: (1) system setup phase, (2) proxy phase (Figure 8), (3) register phase (Figure 9), (4) circling phase (Figure 10), (5) voting phase (Figure 11), and (6) counting phase (Figure 12), as shown in Figure 7.

Step 1. The system first generates the required parameters.

Step 2a. The creator delegates his or her authority to a proxy creator.

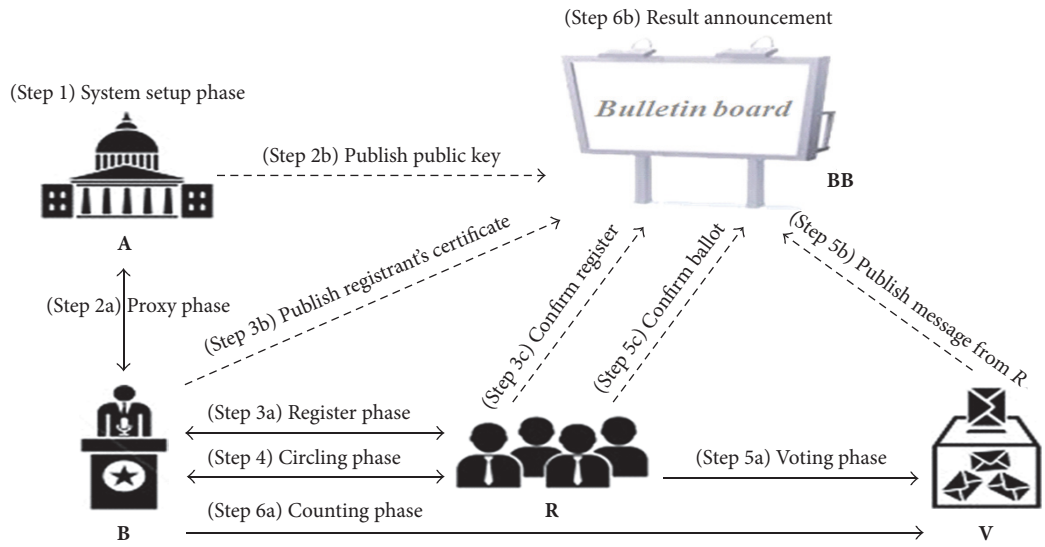


Figure 7: System diagram.

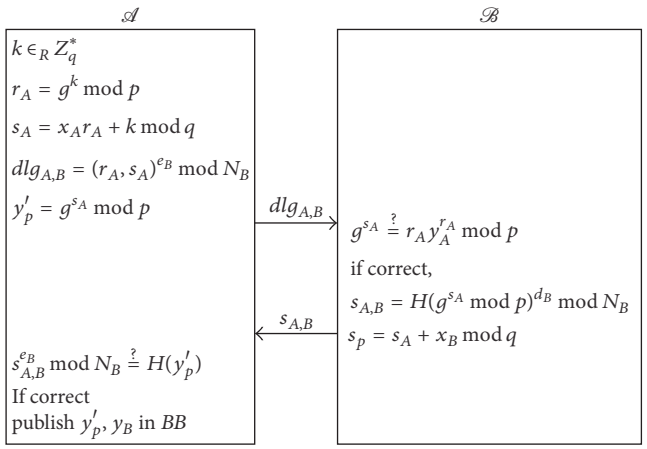


Figure 8: Proxy phase.

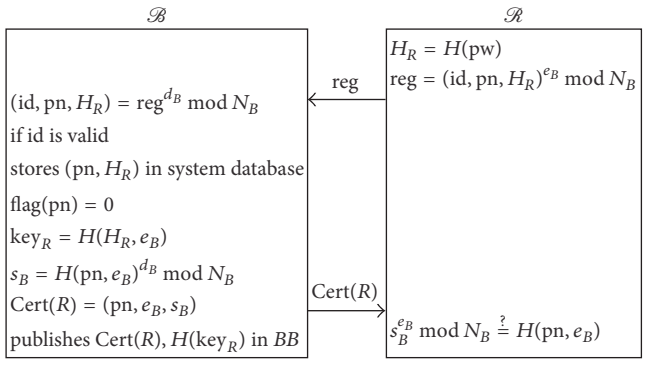


Figure 9: Register phase.

Step 2b. The creator publishes the public key of the proxy creator to the bulletin.

Step 3a. The proxy creator examines whether the registrant is a legal voter; if so, he or she distributes a certificate to the voter.

Step 3b. The proxy creator publishes the certificate to the bulletin.

Step 3c. The voter checks whether he or she has registered successfully via the bulletin.

Step 4. The voter chooses a candidate and receives the signature on it from the proxy creator.

Step 5a. The voter casts his or her ballot and sends it to the voting center.

Step 5b. The voting center publishes a message about the ballot from the voter to the bulletin.

Step 5c. Every voter can confirm whether his or her ballot has been received by the voting center; if not, he or she can resend the ballot.

Step 6a. At the end of the voting period, the proxy creator forwards the decrypting key to the voting canter, and the voting center starts to verify and count the ballots.

Step 6b. The voting center publishes the voting result to the bulletin, where everyone can verify and count all ballots.

5.2. System Process. We assume that the system database already contains an identification list of legal voters and that the bulletin is read-only to all entities except for the authorities.

(1) System Setup Phase

Step 1. Two large primes $p, q \ni q \mid (p-1)$ are chosen.

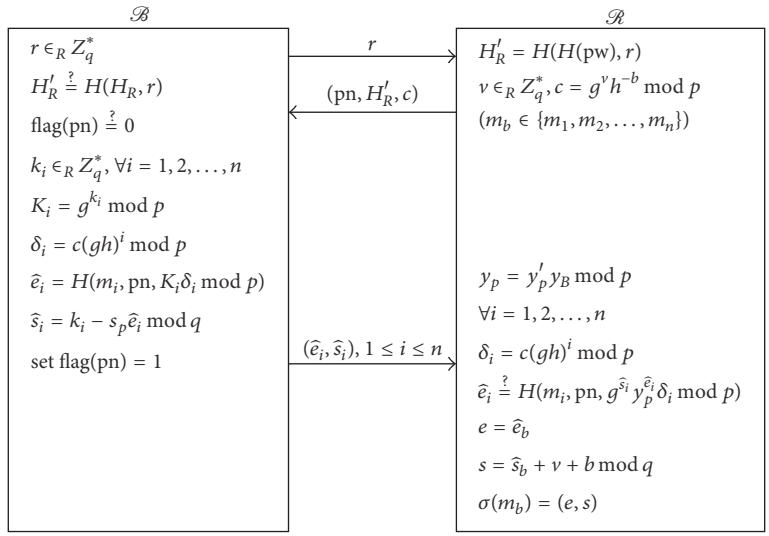


Figure 10: Circling phase.

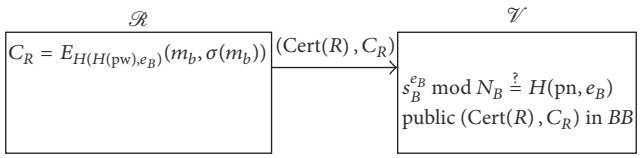


Figure 11: Voting phase.

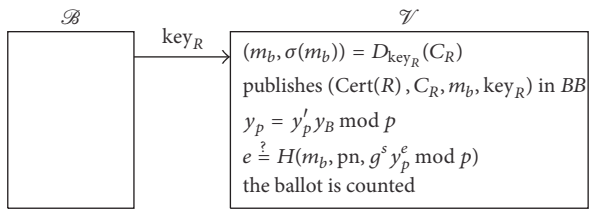


Figure 12: Counting phase.

Step 2. Two generators $g, h \in Z_p^*$, where $\text{Ord}_p g = q$ and $\text{Ord}_p h = q$, are chosen.

Step 3. The creator $\mathscr{A}$ chooses $x_A \in Z_q^*$ and computes $y_A = g^{x_A} \bmod p$.

Step 4. The proxy creator $\mathscr{B}$ chooses $x_B \in Z_q^*$ and computes $y_B = g^{x_B} \bmod p$.

Step 5. $\mathscr{B}$ chooses two large primes p_B, q_B.

Step 6. $\mathscr{B}$ computes $N_B = p_B \times q_B$.

Step 7. $\mathscr{B}$ computes $\phi(N_B) = (p_B - 1)(q_B - 1)$.

Step 8. $\mathscr{B}$ chooses $e_B \ni \text{GCD}(e_B, \phi(N_B)) = 1$.

Step 9. $\mathscr{B}$ computes $d_B = e_B^{-1} \bmod \phi(N_B)$.

(2) Proxy Phase

Step 1. $\mathscr{A}$ randomly chooses $k \in_R Z_q^*$ and computes $r_A = g^k \bmod p$, $s_A = x_A r_A + k \bmod q$, and $y_p' = g^{s_A} \bmod p$.

Step 2. $\mathscr{A}$ encrypts the pair (r_A, s_A) using (e_B, N_B), forwards it to $\mathscr{B}$, and publishes y_p'.

Step 3. $\mathscr{B}$ decrypts (r_A, s_A) using (d_B, N_B) and checks whether $g^{s_A} = r_A y_A^{r_A} \bmod p$ holds. If it does, $\mathscr{B}$ accepts the proxy and computes $s_p = s_A + x_B \bmod q$ as his or her secret proxy signature key.

Step 4. $\mathscr{B}$ generates a signature $s_{A,B} = H(g^{s_A} \bmod p)^{d_B} \bmod N_B$ and forwards it to $\mathscr{A}$.

Step 5. $\mathscr{A}$ checks whether $s_{A,B}^{e_B} = H(y_p') \bmod N_B$ holds. If it does, he or she publishes y_p', y_B to the bulletin.

(3) Register Phase

Step 1. A voter $\mathscr{R}$ picks a pseudoname pn and a password pw, computes $H_R = H(\text{pw})$, encrypts (id, pn, $H(\text{pw})$) using (e_B, N_B), and sends it to a proxy creator.

Step 2. $\mathscr{B}$ decrypts (id, pn, H_R) using (d_B, N_B) and checks whether $\mathscr{R}$ is a legal voter. If so, $\mathscr{B}$ stores (pn, H_R) in the system database, sets flag(pn) = 0, calculates key$_R$ = $H(H_R, e_B)$ and $s_B = H(\text{pn}, e_B)^{d_B} \bmod N_B$, returns Cert($R$) to $\mathscr{R}$, and publishes Cert(R) to the bulletin, where Cert(R) = (pn, e_B, s_B).

Step 3. $\mathscr{R}$ verifies whether $s_B^{e_B} = H(\text{pn}, e_B) \bmod N_B$ is correct. If so, he or she has the right to vote.

(4) Circling Phase

Step 1. $\mathscr{B}$ sends a random number $r \in_R Z_q^*$ to $\mathscr{R}$ after receiving a login request from $\mathscr{R}$.

Step 2. $\mathscr{R}$ computes $H_R' = H(H(\text{pw}), r)$, picks a random number $v \in_R Z_q^*$, calculates $c = g^v h^{-b} \bmod p$ ($m_b \in \{m_1, m_2, \ldots, m_n\}$), and forwards (pn, H_R', c) to $\mathscr{B}$.

Step 3. $\mathscr{B}$ examines whether $H_R' = H(H_R, r)$ is correct. If so, $\mathscr{B}$ checks whether flag(pn) = 0. If it holds, $\mathscr{B}$ chooses $k_i \in_R Z_q^*$, calculates $K_i = g^{k_i} \bmod p$, $\delta_i = c(gh)^i \bmod p$, $\widehat{e}_i = H(m_i, \text{pn}, K_i\delta_i \bmod p)$, and $\widehat{s}_i = k_i - s_p\widehat{e}_i \bmod q$, $\forall i = 1, 2, \ldots, n$, returns $(\widehat{e}_i, \widehat{s}_i)$, $1 \le i \le n$, to $\mathscr{R}$, and sets flag(pn) = 1.

Step 4. $\mathscr{R}$ computes $y_p = y_p' y_B \bmod p$, and, for every $i = 1, 2, \ldots, n$, he or she calculates $\delta_i = c(gh)^i \bmod p$ and checks whether $\widehat{e}_i = H(m_i, \text{pn}, g^{\widehat{s}_i} y_p^{\widehat{e}_i}\delta_i \bmod p)$ is correct. If so, $\mathscr{R}$ computes $s = \widehat{s_b} + v + b \bmod q$ and $e = \widehat{e_b}$. The final signature is $\sigma(m_b) = (e, s)$.

(5) Voting Phase

Step 1. $\mathscr{R}$ calculates $H(H(\text{pw}), e_B)$ and uses it as a symmetric key to encrypt $(m_b, \sigma(m_b))$, produces a cipher C_R, and sends $(\text{Cert}(R), C_R)$ to the voting center.

Step 2. $\mathscr{V}$ first examines whether $s_B^{e_B} = H(\text{pn}, e_B) \bmod N_B$ holds. If so, $\mathscr{V}$ publishes $(\text{Cert}(R), C_R)$ to the bulletin.

Step 3. Every voter can check whether his or her ballot is received by the voting center via the bulletin. If it is not, the voter resends $(\text{Cert}(R), C_R)$.

(6) Counting Phase

Step 1. $\mathscr{B}$ forwards $\text{key}_R = H(H_R, e_B)$ to the voting center.

Step 2. $\mathscr{V}$ decrypts C_R using the symmetric key key_R, publishes $(\text{Cert}(R), C_R, m_b, \text{key}_R)$ to the bulletin and calculates $y_p = y_p' y_B \bmod p$, and verifies whether $e = H(m_b, \text{pn}, g^s y_p^e \bmod p)$ is correct. If so, the signature is valid and the ballot is counted.

Step 3. $\mathscr{V}$ publishes the voting result. Everyone can verify and count the ballots via the bulletin.

6. Security Analysis

In this section, we analyze our protocol according to the security requirements defined in Section 3.

6.1. Signature Scheme

(1) The Proposed Scheme Is Complete. Because the completeness of the signature $\sigma = (e, s)$ depends on the Schnorr signature [28], we only show the completeness of the oblivious signature generated in the signing phase. For any $(\widehat{e}_i, \widehat{s}_i)$, $i = 1, 2, \ldots, n$, we have $H(m_i, g^{\widehat{s}_i} y_p^{\widehat{e}_i}\delta_i \bmod p) = H(m_i, g^{k_i - s_p\widehat{e}_i} g^{s_p\widehat{e}_i}\delta_i \bmod p) = H(m_i, g^{k_i}\delta_i \bmod p) = H(m_i, K_i\delta_i \bmod p) = \widehat{e}_i$. Therefore, the completeness of the proxy signature $\sigma = (e, s)$ is proved.

(2) Proxy Oblivious Signatures Are Unforgeable. Unforgeability can be proved via Definitions 3 and 5 and Theorems 4 and 6.

Definition 3 (DLP). Let p_1 be a large prime, g_1 be a primitive root modulo p, and $y = {g_1}^x \bmod p_1$, where $x, y \in Z_{p_1}^*$. If x can be evaluated from given y, p_1, and g_1, then we say discrete logarithm problem (DLP) can be solved (the probability of solving this problem is denoted as $\Pr(x \mid y, p_1, g_1) = \varepsilon_1$).

Theorem 4 (unforgeability). *In our protocol, if a recipient can forge the signer's signature; then DLP can be solved.*

Proof. In our scheme, assume an adversary tries to evaluate s_A by eavesdropping y_p'. Let RO_1 be a random oracle: input y_p', p, and g to output s_A (i.e., $\text{RO}_1(y_p', p, g) \Rightarrow s_A$). In Definition 3, let $y_p' \leftarrow x$, $p \leftarrow p_1$, and $g \leftarrow g_1$ be input parameters of RO_1 and obtain output s_A. Let $y \leftarrow s_A$; then y is evaluated. Therefore, $\Pr(s_A \mid y_p', p, g) \le \Pr(y \mid x, p_1, g_1) = \varepsilon_1$, which means the *discrete logarithm problem* can be solved if RO_1 exists. □

Definition 5 (DLP under known plaintext attack). Let p_1 be a large prime, g_1 be a primitive root modulo p_1, and $y_i = g^{x_i} \bmod p_1$, where $x_i, y_i \in Z_{p_1}^*$. If x_{n+1} can be evaluated from given p_1, g_1, (x_i, y_i), and y_{n+1}, $i = 1, 2, \ldots, n$, then we say DLP under known plaintext attack can be solved (the probability of solving this problem is denoted as $\Pr(x_{n+1} \mid (x_i, y_i), y_{n+1}, p_1, g_1) = \varepsilon_2$).

Theorem 6 (unforgeability under replay attack). *In our protocol, if a recipient can forge the signer's signature from given n pairs* $(y_P'^{(i)}, s_A^{(i)})$, $i = 1, 2, \ldots, n$, *then DLP under known plaintext attack can be solved.*

Proof. In our scheme, assume an adversary tries to evaluate $s_A^{(i+1)}$ from eavesdropped $(y_P'^{(i)}, s_A^{(i)})$ and $y_P'^{(n+1)}$, $i = 1, 2, \ldots, n$. Let RO_2 be a random oracle: input $(y_P'^{(i)}, s_A^{(i)})$, $y_P'^{(n+1)}$, p, g to output $s_A^{(i+1)}$ (i.e., $\text{RO}_2((y_P'^{(i)}, s_A^{(i)}), y_P'^{(n+1)}, p, g) \Rightarrow s_A^{(i+1)}$). In Definition 5, let $(y_P'^{(i)}, s_A^{(i)}) \leftarrow (x_i, y_i)$, $y_P'^{(n+1)} \leftarrow y_{n+1}$, $p \leftarrow p_1$, and $g \leftarrow g_1$ be input parameters of RO_2 and obtain output $s_A^{(i+1)}$. Let $x_{n+1} \leftarrow s_A^{(i+1)}$; then x_{n+1} is evaluated. Therefore, $\Pr(s_A^{(i+1)} \mid (y_P'^{(i)}, s_A^{(i)}), y_P'^{(n+1)}, p, g) \le \Pr(x_{n+1} \mid (x_i, y_i), y_{n+1}, p_1, g_1) = \varepsilon_2$, which means the *DLP under known plaintext attack* can be solved if RO_2 exists. □

(3) Proxy Oblivious Signatures Are Unlinkable. In the signing phase, because $\mathscr{B}$ receives the disguised selection c of the

Table 1: Computation cost comparison.

Scheme	Original signer	(Proxy) signer	Receiver	Verifier
Yang and Liang [19]	T_{ex}	$4T_{ex}$	$2T_{ex}$	$3T_{ex}$
Chen [10]	-	$3nT_{ex}$	$(2n+10)T_{ex}$	$8T_{ex}$
Tso et al. [11]	-	$2nT_{ex}$	$(2n+2)T_{ex}$	$2T_{ex}$
Proposed	$2T_{ex}$	$(n+2)T_{ex}{}^{*}$	$(2n+2)T_{ex}$	$2T_{ex}$

message m_b, he or she cannot find out the selection of $\mathscr{R}$. Moreover, because the blinding factor v is required during the conversion of the oblivious signature $\widehat{s_b}$, where v is known only by $\mathscr{R}$, $\mathscr{B}$ cannot make a linkage between the signature and $\mathscr{R}$'s selection. Thus, we can conclude that $\mathscr{B}$ can discover neither the message nor the proxy signature associated with the protocol after the signature is verified.

(4) Proxy Oblivious Signatures Are Undeniable. Owing to the unforgeability of the proxy oblivious signature, $\mathscr{B}$ cannot repudiate any proxy signature produced by him or her.

(5) Proxy Oblivious Signatures Are Verifiable. In the verification phase, $\mathscr{V}$ checks whether the equation $e = H(m_b, g^s y_p^e \bmod p)$ holds, because $g^s y_p^e = g^{\widehat{s_b}+v+b} y_p^{\widehat{e_b}} = g^{k_b - s_p\widehat{e_b}+v+b} g^{s_p\widehat{e_b}} = g^{k_b+v+b} = g^{k_b} g^v h^{-b} g^b h^b = g^{k_b} c(gh)^b = K_b\delta_b \bmod p$, and, thus, $H(m_b, g^s y_p^e \bmod p) = H(m_b, K_b\delta_b \bmod p) = \widehat{e_b} = e$. Therefore, the signature can be verified by all verifiers.

(6) Generated Signatures Are Distinguishable. Evidently, by using different congruences to check the validity of the original signatures and the proxy signatures, everyone can easily distinguish the proxy signature from a normal signature.

(7) The Proposed Scheme Provides Ambiguity of the Message Selected by $\mathscr{R}$; That Is, It Provides Perfect Security for $\mathscr{R}$. If an attacker $\mathscr{A}_1$ takes the place of $\mathscr{B}$, $\mathscr{A}_1$ needs to obtain b, where $c = g^v h^{-b} \bmod p$, $v \in_R Z_q^*$, because we know that, for every b of any c, there exists v_i such that $c = g^v h^{-b} = g^{v_i} h^{-i} \bmod p$, for $i = 1, 2, \ldots, n$. Finally, we conclude that the probability of $\mathscr{A}_1$ getting the correct b is $1/n$, which achieves theoretical security.

6.2. Voting System

(1) Delegation from Creator Is Complete. In the proxy phase, $\mathscr{B}$ signs the proxy public key y_p' using his or her private key x_B and passes the signature $s_{A,B}$ to $\mathscr{A}$. Therefore, $\mathscr{B}$ cannot deny the fact that he or she has accepted the delegation from $\mathscr{A}$.

(2) Eligibility. In the register phase, $\mathscr{B}$ checks whether $\mathscr{R}$ is a legal voter by his or her id. Therefore, only legal voters can get the certificate CERT(R) from $\mathscr{B}$ and participate in the voting event.

(3) Nonduplication. In the circling phase, $\mathscr{B}$ checks whether flag(pn) = 0. If it holds, it means that pn has not voted yet; otherwise, the process will be terminated. Consequently, every legal voter can cast at most one ballot.

Table 2: Communication cost comparison.

Scheme	$A \to B$	$B \to R$	$R \to B$	$R \to V$
Yang and Liang [19]	$l_q + l_H$	$l_p + l_q + l_H$	l_q	$l_q + 2l_H$
Chen [10]	-	$3nl_p + nl_q$	l_q	$7l_p + l_q + l_H$
Tso et al. [11]	-	$n(l_q + l_H)$	l_p	$l_q + l_H$
Proposed	$l_p + l_q$	$n(l_q + l_H)$	l_p	$l_q + l_H$

(4) Fairness. In the voting phase, $\mathscr{R}$ encrypts his or her ballot using the symmetric key $H(H(\text{pw}), e_B)$ and sends it to the voting center. Therefore, no one can learn the current voting situation before the voting period is over.

(5) Accuracy. The validity of all ballots can be verified using the examining equation $e = H(m_b, \text{pn}, g^s y_p^e \bmod p)$.

(6) Verifiability. In the voting phase, every voter can check his or her own ballot via the bulletin, and the ballot cannot be modified by anyone. Furthermore, in the counting phase, every voter can verify and count all other voters' ballots on the bulletin.

(7) Privacy. In the circling phase, owing to the ambiguity of the proxy oblivious signature, no one can learn a voter's selection from $c = g^v h^{-b} \bmod p$. In the voting phase, $\mathscr{R}$ encrypts his or her ballot using the key $H(H(\text{pw}), e_B)$ before casting it to $\mathscr{V}$, and, therefore, no one can obtain his or her plain ballot. Finally, in the counting phase, because every selected candidate on the bulletin corresponds to a pseudoname pn, no one can discover which person has made the selection.

7. Comparison

This section presents a comparison of our scheme against other related schemes, including blind signature [8], proxy signature [13], oblivious signature [10, 11], and proxy blind signature [19]. Tables 1, 2, and 3 present the computation cost, communication cost, and ability comparison, respectively. Because modular exponentiation is the most significant computational operation, we denote its time cost as "T_{ex}" and ignore the other operations in the schemes.

Compared with other related schemes, our scheme provides the most abilities with a low increment in computation cost. Furthermore, the communication cost is no higher than that of other oblivious signature schemes.

Table 3: Ability comparison.

Scheme	Blindness	Ambiguity	Proxy ability
Nayak et al. [8]	√		
Mambo et al. [13]			√
Yang and Liang [19]	√		√
Chen [10]	√	√	
Tso et al. [11]	√	√	
Proposed	√	√	√

Table 4: Computation time (milliseconds).

Phase	PP		RP		CiP		VP		CoP
Role	A	B	R	B	R	B	R	V	V
Time	42.4	31.5	22.85	19.55	31.2	31	19.8	10.35	20.25

In the proxy phase, $\mathscr{B}$ processes $2T_{ex}$ to examine whether $g^{s_A} = ry_A^r \bmod p$ holds (see asterisk in Table 1). In the signing phase, $\mathscr{B}$ processes nT_{ex} to calculate $K_i = g^{k_i} \bmod p$ for $i = 1, 2, \ldots, n$. For $\delta_i = c(gh)^i \bmod p$, $\mathscr{B}$ may compute δ_i by letting $\delta_0 = c$ and generate $\delta_i = \delta_{i-1}(gh) \bmod p$ for $i = 1, 2, \ldots, n$. Consequently, the computation cost is n modular multiplication rather than nT_{ex}.

8. Implementation

This section demonstrates an application of an anonymous mobile electronic voting with proxy signer implemented on smartphones. Figure 13 shows a flowchart of this system. In this system, it is assumed that any signer who is asked for his or her public key responds with his or her real public key immediately and that the requester would receive this public key at once. Tables 4, 5, and 6 present the average computation times of each role in each phase, the average communication times of each communication direction in each phase, and the average computation and communication times in each phase, respectively, where PP, RP, CiP, VP, and CoP stand for proxy phase, register phase, circling phase, voting phase, and counting phase. The steps in this system are as follows.

Step 1. On the main page (Figure 14(a)), users are allowed to select a role and enter a pseudoname (Figure 14(b)). An original signer first runs the procedure and waits for proxy signers (Figure 14(c)).

Step 2. A proxy signer runs Step 1 and selects a detected original signer to request delegation (Figure 14(d)).

Step 3. The original signer selects the proxy signer on the request list to process the delegation (Figure 14(e)).

Step 4. After verifying the correctness of the delegation, the proxy signer sets the voting issue (Figure 14(f)) and starts waiting for verifiers (Figure 14(g)).

Step 5. A verifier runs Step 1 and waits for a proxy signer to send him a voting event (Figure 14(h)).

Step 6. The proxy signer selects a verifier to send the voting event (Figure 14(i)) and starts holding the event (Figure 14(j)).

Step 7. A receiver selects "Voter" on the main page and chooses a detected voting event (Figure 14(k)).

Step 8. The receiver votes for a candidate and presses the "Vote" button (Figure 14(l)) to send his or her vote to the proxy signer. Then, he or she receives a proxy oblivious signature and sends the extracted signature to the verifier by pressing the "Send" button (Figure 14(m)).

Step 9. The verifier keeps collecting votes (Figure 14(n)) until the proxy signer ends the voting event by pressing the "End Event" button shown in Figure 14(j).

Step 10. After the end of the voting event, the verifier starts verifying the collected votes and displays the voting result (Figure 14(o)).

9. Conclusion

We first construct 1-out-of-n proxy oblivious signature schemes of proxy-unprotected and proxy-protected types and discuss their security requirements. The proposed schemes combine the advantages of a proxy signature and an oblivious signature and satisfy the security properties of both signatures, including completeness, unforgeability, unlinkability, undeniability, verifiability, distinguishability, and ambiguity. Compared with related schemes, our schemes provide extra proxy ability and perform well in terms of both complexity and usability. Finally, an anonymous proxy electronic voting application is implemented on smartphones based on the proposed scheme.

Nomenclature

Notation

$\mathscr{A}$:	The original signer
$\mathscr{B}$:	The proxy signer
$\mathscr{R}$:	The recipient
$\mathscr{V}$:	The verifier
p, q:	Two large prime numbers such that $q \mid p - 1$
g, h:	Two elements of Z_p^* of the same order q
Z_n^*:	A set of the integers in $\{1, 2, \ldots, n-1\}$ that are coprime to n
$\mathrm{Ord}_x y$:	The order of y modulo x
$x \in_R Z_q^*$:	A chosen random number x in Z_q^*
x_A:	A's private key
y_A:	A's public key
s_p:	The signing key
n:	The number of messages
m_i:	The ith message
b:	The value of the subscript of the selected message m_b
σ:	The signature on m_b
$H(\cdot)$:	A public one-way hash function.

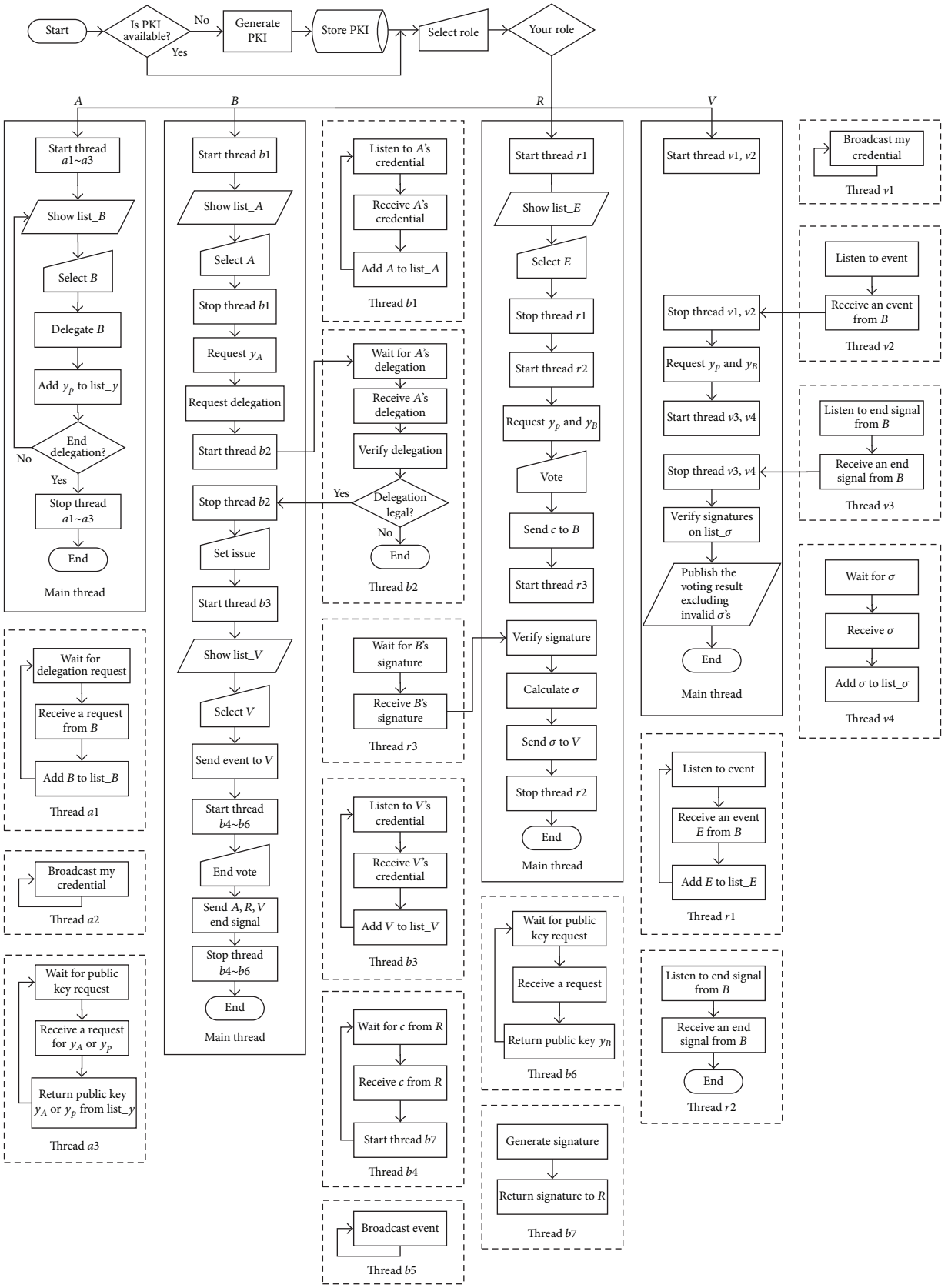
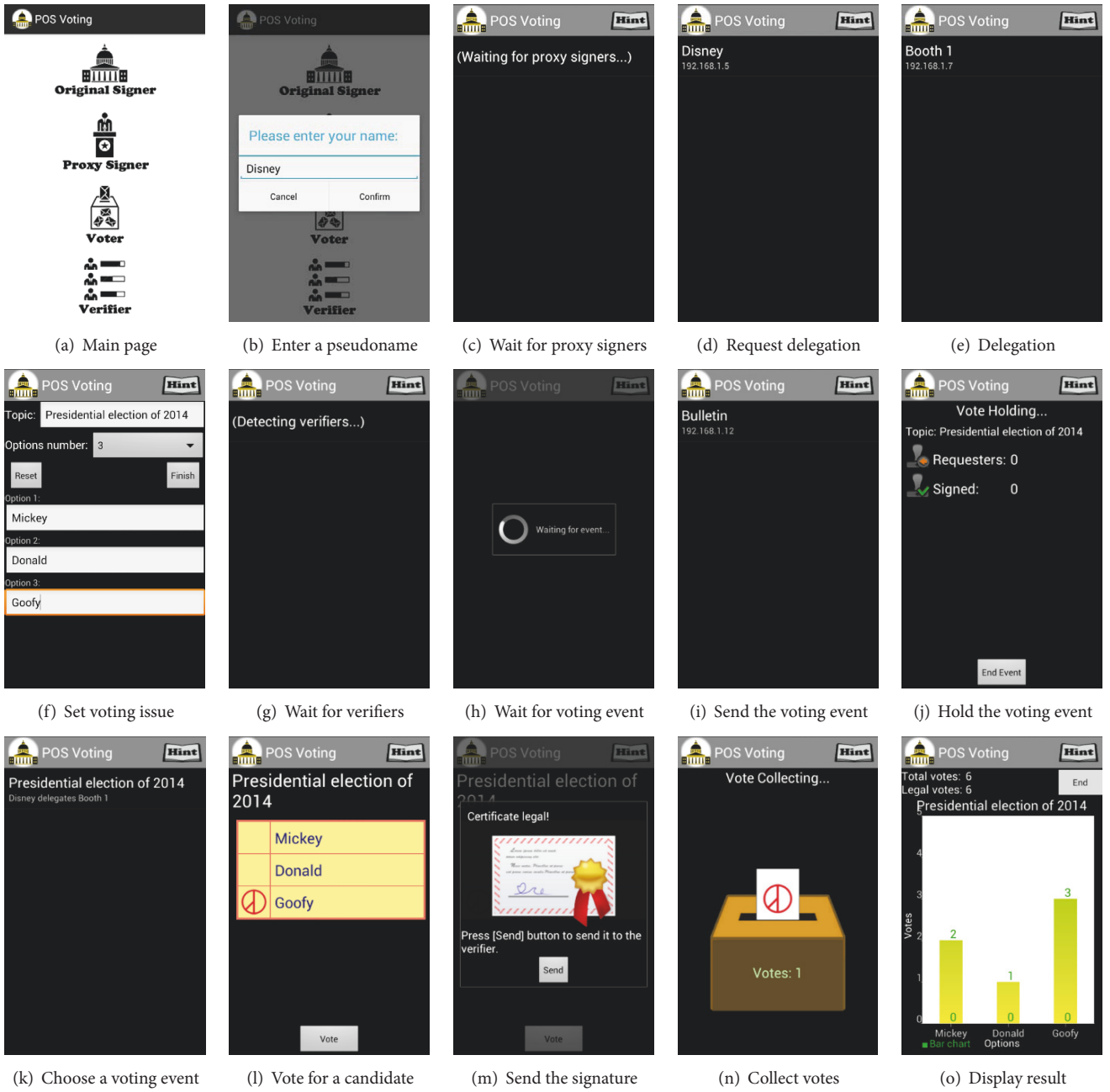


Figure 13: Flowchart of voting application.

Table 5: Communication time (milliseconds).

Phase	PP		RP		CiP			VP	CoP
Direction	$A \to B$	$B \to A$	$R \to B$	$B \to R$	$B \to R$	$R \to B$	$B \to R$	$R \to V$	$B \to V$
Time	61.05	81.65	32.1	42.8	44.45	38.75	47.2	44.7	39.6



(a) Main page (b) Enter a pseudoname (c) Wait for proxy signers (d) Request delegation (e) Delegation

(f) Set voting issue (g) Wait for verifiers (h) Wait for voting event (i) Send the voting event (j) Hold the voting event

(k) Choose a voting event (l) Vote for a candidate (m) Send the signature (n) Collect votes (o) Display result

Figure 14: Data transfer.

Symbols Used in Performance Analysis

$\mathscr{A}$: The original signer
$\mathscr{B}$: The proxy signer
$\mathscr{R}$: The recipient
$\mathscr{V}$: The verifier
p, q: Two large prime numbers such that $q \mid p - 1$
g, h: Two elements of Z_p^* of the same order q
x_A: A's private key
y_A: A's public key
s_p: The signing key

Table 6: Implementation time (milliseconds).

Phase	PP	RP	CiP	VP	CoP	Total
Computation time	73.9	42.4	62.2	30.15	20.25	228.9
Communication time	142.7	74.9	130.4	44.7	39.6	432.3

n: The number of messages
m_i: The ith message
b: The value of the subscript of the selected message m_b
n: Number of candidate messages in oblivious signature scheme (e.g., 10)
T_{ex}: Time cost of a modular exponentiation operation
l_q: Length of the parameter q (e.g., 128 bits)
l_p: Length of the parameter p (e.g., 1024 bits)
l_H: Length of the output of hash function (e.g., 128 bits)
σ: The signature on m_b
$H(\cdot)$: A public one-way hash function.

Disclosure

This article does not contain any studies with human participants or animals performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is partially supported and funded by the Ministry of Science and Technology under Grant MOST 104-2221-E-182-012 and by the CGMH project under Grant BMRPB46.

References

[1] J. Modic, R. Trapero, A. Taha, J. Luna, M. Stopar, and N. Suri, "Novel efficient techniques for real-time cloud security assessment," *Computers & Security*, vol. 62, pp. 1–18, 2016.

[2] A. V. Uzunov, "A survey of security solutions for distributed publish/subscribe systems," *Computers & Security*, vol. 61, pp. 94–129, 2016.

[3] A. V. Uzunov, E. B. Fernandez, and K. Falkner, "Security solution frames and security patterns for authorization in distributed, collaborative systems," *Computers & Security*, vol. 55, pp. 193–234, 2015.

[4] A. Castiglione, F. Palmieri, U. Fiore, A. Castiglione, and A. De Santis, "Modeling energy-efficient secure communications in multi-mode wireless mobile devices," *Journal of Computer and System Sciences*, vol. 81, no. 8, pp. 1464–1478, 2015.

[5] T. Li, Z. Liu, J. Li, C. Jia, and K.-C. Li, "CDPS: a cryptographic data publishing system," *Journal of Computer and System Sciences*, vol. 89, pp. 80–91, 2016.

[6] Q. Shi, N. Zhang, and M. Merabti, "Fair signature exchange via delegation on ubiquitous networks," *Journal of Computer and System Sciences*, vol. 81, no. 4, pp. 615–631, 2015.

[7] D. Chaum, "Blind signatures for untraceable payments," *Crypto*, vol. 82, 1983.

[8] S. K. Nayak, B. Majhi, and S. Mohanty, "An ECDLP based untraceable blind signature scheme," in *Proceedings of the Circuits, Power and Computing Technologies (ICCPCT), International Conference on. IEEE*, pp. 829–834, India, March 2013.

[9] M. O. Rabin, "How to exchange secrets by oblivious transfer," *IACR Cryptology ePrint Archive*, vol. 2005, p. 187, 2005.

[10] L. Chen, "Oblivious signatures," in *Computer Security–ESORICS 94, Lecture Notes in Computer Science 875*, vol. 875 of *Lecture Notes in Computer Science*, pp. 161–172, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.

[11] R. Tso, T. Okamoto, and E. Okamoto, "1-out-of-n oblivious signatures," *Information Security Practice and Experience*, vol. 4991, pp. 45–55, 2008, Springer Berlin Heidelberg.

[12] J.-S. Chou, "A novel k-out-of-n oblivious transfer protocol from bilinear pairing," *Advances in Multimedia*, vol. 2012, Article ID 630610, 2012.

[13] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, no. 9, pp. 1338–1353, 1996.

[14] B. T. Lau, "Proxy signature schemes," in *Proceedings of the 2006 1st IEEE Conference on Industrial Electronics and Applications, ICIEA 2006*, Singapore, May 2006.

[15] H. Wang and R. Yan, "A code-based multiple grade proxy signature scheme," in *Proceedings of the 2013 8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 3PGCIC 2013*, pp. 559–562, France, October 2013.

[16] W. D. Lin and J. K. Jan, "A security personal learning tools using a proxy blind signature scheme," in *Proceedings of the Intl Conference on Chinese Language Computing*, pp. 273–277, 2000.

[17] A. Z. Tan, Z. Liu, and C. Tang, "Digital proxy blind signature schemes based on DLP and ECDLP," *MM Research Preprints*, vol. 21, no. 7, pp. 212–217, 2002.

[18] S. Lal and A. K. Awasthi, "Proxy Blind Signature Scheme," *Journal of Information Science and Engineering*, vol. 72, 2003.

[19] F.-Y. Yang and L.-R. Liang, "A proxy partially blind signature scheme with proxy revocation," *Journal of Ambient Intelligence and Humanized Computing*, vol. 4, no. 2, pp. 255–263, 2013.

[20] I. Ray and N. Narasimhamurthi, "An anonymous electronic voting protocol for voting over the internet," *Advanced Issues of E-Commerce and Web-Based Information Systems, Third International Workshop on IEEE*, pp. 188–190, 2001.

[21] H. Pan, E. Hou, and N. Ansari, "RE-NOTE: An E-voting scheme based on ring signature and clash attack protection," in *Proceedings of the 2013 IEEE Global Communications Conference, GLOBECOM 2013*, pp. 867–871, USA, December 2013.

[22] B. Zwattendorfer, C. Hillebold, and P. Teufl, "Secure and privacy-preserving proxy voting system," in *Proceedings of the 2013 IEEE 10th International Conference on e-Business Engineering, ICEBE 2013*, pp. 472–477, UK, September 2013.

[23] S. Kardaş, M. S. Kiraz, M. A. Bingöl, and F. Birinci, "Norwegian internet voting protocol revisited: ballot box and receipt generator are allowed to collude," *Security and Communication Networks*, vol. 9, no. 18, pp. 5051–5063, 2016.

[24] O. Kulyk, S. Neumann, K. Marky, J. Budurushi, and M. Volkamer, "Coercion-resistant proxy voting," in *Proceedings of the 31st International Conference on ICT Systems Security and Privacy Protection*, vol. 471, pp. 3–16, IFIP SEC, 2016.

[25] O. Kulyk, K. Marky, S. Neumann, and M. Volkamer, "Introducing proxy voting to Helios," in *Proceedings of the 11th International Conference on Availability, Reliability and Security (ARES)*, pp. 98–106, Austria, September 2016.

[26] B. Adida, "Helios: Web-based open-audit voting," *USENIX Security Symposium*, vol. 17, pp. 335–348, 2008.

[27] G. Cohensius, S. Mannor, R. Meir, E. Meirom, and A. Orda, "Voting for better outcomes," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 858–866, São Paulo, Brazil, 2017.

[28] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[29] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, November 1993.

[30] C. Shin-Yan, W. Tsung-Ju, and C. Jiun-Ming, "1-out-of-n Proxy Oblivious Signature Schemes and its e-voting application," in *Proceedings of the International Conference on Electronics Systems and Information Technology (ICESIT-15)*, pp. 10–14, 2015.

[31] E. Aljarrah, H. Elrehail, and B. Aababneh, "E-voting in Jordan: Assessing readiness and developing a system," *Computers in Human Behavior*, vol. 63, pp. 860–867, 2016.

[32] D. A. López García, "A flexible e-voting scheme for debate tools," *Computers & Security*, vol. 56, pp. 50–62, 2016.

[33] K. Vassil, M. Solvak, P. Vinkel, A. H. Trechsel, and R. M. Alvarez, "The diffusion of internet voting. Usage patterns of internet voting in Estonia between 2005 and 2015," *Government Information Quarterly*, vol. 33, no. 3, pp. 453–459, 2016.

[34] J. Dossogne and L. Frédéric, "Blinded additively homomorphic encryption schemes for self-tallying voting," *Journal of Information Security and Applications*, vol. 22, pp. 40–53, 2015.

[35] N. András, A. Márta, and S. Péter, "Could on-line voting boost desire to vote? – Technology acceptance perceptions of young Hungarian citizens," *Government Information Quarterly*, vol. 33, no. 4, pp. 705–714, 2016.

[36] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," in *Advances in Cryptology*, vol. 196 of *Lecture Notes in Computer Science*, pp. 10–18, Springer, Berlin, Germany, 1985.

[37] S.-Y. Chiou, Z. Ying, and J. Liu, "Improvement of a privacy authentication scheme based on cloud for medical environment," *Journal of Medical Systems*, vol. 40, no. 4, article no. 101, pp. 1–15, 2016.

[38] S.-Y. Chiou, "Common friends discovery for multiple parties with friendship ownership and replay-attack resistance in mobile social networks," *Wireless Networks*, pp. 1–15, 2016.

[39] R. Anane, R. Freeland, and G. Theodoropoulos, "E-voting requirements and implementation," in *Proceedings of the 9th IEEE International Conference on E-Commerce Technology; The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services, CEC/EEE 2007*, pp. 382–389, Japan, July 2007.